On-Orbit Optical Camera Self-Calibration for Cislunar Explorers

Michael Wang, Kyle Doyle, and Mason Peck

Cornell University, Ithaca NY, 14853

I. Abstract

This manuscript describes a method for estimating an optical camera's intrinsic parameters as well as lens distortion parameters from two or more images with relative point correspondences. The method uses a nonlinear least squares algorithm with optional constraints and requires no inertial information such as star positions. This is known as self-calibration, since the method relies on relative rather than inertial point correspondences for calibration. This approach is applied to the specific problem of on-orbit camera calibration for the optical navigation subsystem of Cislunar Explorers; however, it can be extended to applications beyond spacecraft imaging such as calibration of general stereo-vision systems. Through simulated images, the algorithm in this manuscript demonstrates fast convergence from large initial deviations in intrinsic and lens distortion parameters. Sensitivity analysis using image noise is conducted to test the robustness of the algorithm.

II. Introduction

Cislunar Explorers is a CubeSat flight program led by Cornell University, consisting of two spacecrafts to be launched on SLS/EM-1 in 2020. The optical navigation subsystem involves work on image processing and camera calibration as well as trajectory and attitude estimation. Computer vision algorithms extract the angular separations among the Sun, Moon, and Earth as well as their apparent mean radii as measured by onboard cameras. These measurements are compared to Ephemeris and fed into trajectory and attitude estimators to obtain spacecraft position, velocity, attitude, and angular velocity estimates. This manuscript focuses on the camera calibration problem of the optical navigation subsystem.

The system is composed entirely of commercial off-the-shelf (COTS) cameras, which necessitates camera calibration to identify intrinsic parameters and correct for lens distortion. These parameters are needed to accurately calculate rays from the camera origin to specific inertial points and landmarks, which are inputs to the computer vision algorithms for extracting angular separations and mean radii of celestial bodies. In addition, due to intense vibrations from the launch vehicle as well as extreme environmental conditions in cislunar space, regular on-orbit camera calibrations are required. Furthermore, due to the spin-stabilized configurations of the spacecrafts,

quiescent star field images cannot be obtained. As a result, self-calibration using brighter subjects such as surface features of the Moon and the Earth is required.

III. LITERATURE REVIEW

Hartley introduces methods for self-calibration to obtain intrinsic camera parameters assuming pure-rotation motion and pin-hole camera without lens distortion [1]. The pure-rotation motion assumption is valid for spacecraft imaging due to the vast separation between the subject and the camera compared to the lever arm of the camera origin relative to the center of rotation, which is the center of mass of a spacecraft. Methods include a non-iterative algorithm using Cholesky factorization and an iterative algorithm using Levenberg-Marquardt nonlinear least squares optimization. The Levenberg-Marquardt algorithm obtains consistently better results than the non-iterative algorithm at the expense of slightly higher computational overhead.

Wang et. al. introduce an on-orbit batched and sequential estimation algorithm for estimating camera installation angles as well as intrinsic parameters and lens distortion parameters using reference stars as control points [2]. Christian et. al. introduce a method for determining camera intrinsic parameters as well as lens distortion parameters using Levenberg-Marquardt algorithm on an ensemble of star field images with known inertial star positions [3]. The algorithm attempts to minimize the nonlinear least squares error between observed star centroid locations and the expected star centroid locations, computed from a star catalog and camera intrinsic parameters and lens distortion parameters. Our approach differs fundamentally from [3] in that no inertial information is needed, hence the self-calibration. We note that the method in [3] is a standard way of conducting on-orbit calibration; however, Cislunar Explorers' unique mission profile demands a more specialized algorithm, which is provided in this manuscript.

Although not discussed in this manuscript, computer vision algorithms are needed to localize the centroids as well as measure apparent radii of celestial bodies. Mortari et. al. introduce novel algorithms to estimate observer-to-body relative position in inertial coordinates assuming the body is a triaxial ellipsoid [4]. Christian et. al. introduce on-board image processing algorithms for autonomous navigation without Earth communication [5], [6]. In addition, estimation algorithms are needed for feedback control of the spacecraft. Hunter et. al. introduce a method for autonomously recovering spacecraft position, velocity, and time using measurements of celestial bodies from onboard cameras as well as a timer [7], [8]. Crassidis et. al. develop an attitude Multiplicative Extended Kalman Filter (MEKF) using inertial attitude measurements such as a star tracker and an IMU [9].

IV. Camera Model

The basic pinhole model assumes that a light ray from a point in space passing through an ideal lens does not distort. Using similar triangles, the perspective projection equations can be derived:

$$x = f\frac{X}{Z} \qquad y = f\frac{Y}{Z} \tag{1}$$

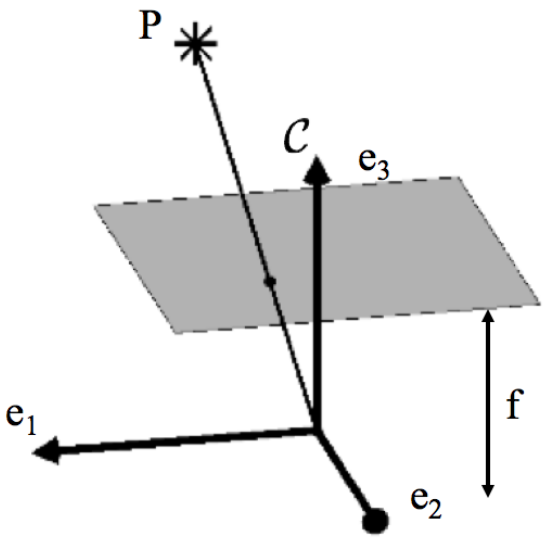

Figure 1: Pinhole Camera Model

where P is a point expressed in camera frame coordinates, $P = \begin{bmatrix} X & Y & Z \end{bmatrix}_C^T$, f is the focal length, and x and y are the image coordinates of P. Equation 1 is expressed in the virtual plane, as seen in Figure 1, rather than the image plane, which is located at $z = -f$. Perspective projection transforms a vector in R^3 to a vector in R^2. This indicates that any point on the ray will project to the same point on the virtual plane and that inverting perspective projection is an ill-posed problem.

The pinhole model is not accurate enough for cameras, especially small COTS cameras such as Raspberry Pi Camera Module v2, due to lens distortion. Lens distortion can be modeled as a nonlinear transformation, d, from distorted image coordinates, $\begin{bmatrix} x_d & y_d \end{bmatrix}^T$, to undistorted image coordinates, $\begin{bmatrix} x & y \end{bmatrix}^T$, assuming $f = 1$ from Equation 1 [10]:

$$\begin{bmatrix} x \\ y \end{bmatrix} = d\left(\begin{bmatrix} x_d \\ y_d \end{bmatrix}\right) = \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right)\begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2p_1 x_d y_d + p_2\left(r^2 + 2x_d^2\right) \\ p_1\left(r^2 + 2y_d^2\right) + 2p_2 x_d y_d \end{bmatrix} \tag{2}$$

where

$$r^2 = x_d^2 + y_d^2$$

k_1, k_2, and k_3 are radial distortion parameters, and p_1 and p_2 are tangential distortion parameters. The choice of $f = 1$ is arbitrary at this step since we have not considered the camera intrinsic

parameters yet. Note, however, the choice of f at this step must be consistent throughout implementation. Also, note that there is a similar model in computer vision literature that transforms undistorted image coordinates to distorted image coordinates; however, they can be chosen depending on the problem at hand [18]. To invert Equation 2, a nonlinear least squares algorithm, such as Levenberg-Marquardt, can be used.

To convert from image coordinates to pixel coordinates, we use the calibration matrix, which is a function of camera intrinsic parameters:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3}$$

where f_x is the focal length scaling in camera x-axis, f_y is the focal length scaling in camera y-axis, s is the skew factor between the camera x and y axes, and $\begin{bmatrix} u_0 & v_0 \end{bmatrix}^T$ is the principal point in pixel coordinates. The upper triangular structure of the calibration matrix indicates that Equation 3 can be easily inverted. In summary, the camera model is governed by 10 parameters (5 intrinsic parameters and 5 lens distortion parameters).

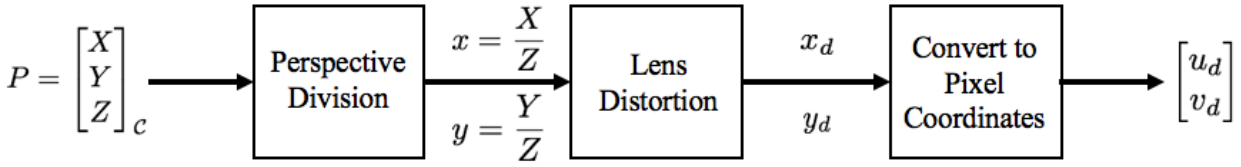

Figure 2: Image Formation Model

V. Problem Statement

The key to the algorithm is the pure-rotation motion assumption, which can be expressed as follows. Given the position vector of an inertial point relative to the camera origin, $\boldsymbol{r}_{P/C}$, and the position vector of the camera origin relative of the center of rotation of the spacecraft, $\boldsymbol{r}_{C/G}$,

$$\left\|\boldsymbol{r}_{P/C}\right\|_2 >> \left\|\boldsymbol{r}_{C/G}\right\|_2 \tag{4}$$

for all time. This greatly simplifies the math for self-calibration, since the relation between the camera frame at time a and time b can be well-approximated using only a rotation matrix ${}^{C_b}R^{C_a}$ that transforms a vector in C_a coordinates to a vector in C_b coordinates.

Given a measurement of an inertial point P at time a, $\begin{bmatrix} u_{da} & v_{da} \end{bmatrix}^T$, and another measurement of the same inertial point P at time b, $\begin{bmatrix} u_{db} & v_{db} \end{bmatrix}^T$, we can find their virtual plane coordinates by

inverting Equation 3:

$$\begin{bmatrix} x_{da} \\ y_{da} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u_{da} \\ v_{da} \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x_{db} \\ y_{db} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u_{db} \\ v_{db} \\ 1 \end{bmatrix} \tag{5}$$

We can then undistort Equation 5 using Equation 2:

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = d\left(\begin{bmatrix} x_{da} \\ y_{da} \end{bmatrix} \right) \qquad \begin{bmatrix} x_b \\ y_b \end{bmatrix} = d\left(\begin{bmatrix} x_{db} \\ y_{db} \end{bmatrix} \right) \tag{6}$$

As shown by Equation 1, the consequence of perspective projection is that we cannot back out the inertial position of P from virtual plane coordinates. We can only determine the position of P up to a scaling factor, i. e. the ray. From the pure-rotation motion assumption, we know there is a point on ray a that is related to a point on ray b via the rotation matrix ${}^{C_b}R^{C_a}$. Therefore, we can convert an equivalence into an equality using scaling factors:

$$\begin{bmatrix} \lambda_b x_b \\ \lambda_b y_b \\ \lambda_b \end{bmatrix} = {}^{C_b}R^{C_a} \begin{bmatrix} \lambda_a x_a \\ \lambda_a y_a \\ \lambda_a \end{bmatrix} \tag{7}$$

with

$${}^{C_b}R^{C_a} = \begin{bmatrix} \boldsymbol{r}_1^T \\ \boldsymbol{r}_2^T \\ \boldsymbol{r}_3^T \end{bmatrix} \tag{8}$$

Note that Equation 8 represents relative attitude if we set C_a to be the "inertial frame". Since self-calibration relies only on relative information, only the attitude of C_b relative to C_a is needed for the algorithm. ${}^{C_b}R^{C_a}$ can be estimated using a MEKF with IMU [9]. To account for estimation errors, we can add variables for attitude correction for each iteration of the algorithm, as shown in [3]:

$${}^{C_b}R^{C_a}{}^{(k+1)} = \delta R^{(k)} {}^{C_b}R^{C_a}{}^{(k)} \tag{9}$$

where ${}^{C_b}R^{C_a}{}^{(k+1)}$ is the rotation matrix at the $(k+1)^{th}$ iteration. Using small angle approximations, the attitude correction can be expressed as:

$$\delta R = \begin{bmatrix} 1 & \delta r_3 & -\delta r_2 \\ -\delta r_3 & 1 & \delta r_1 \\ \delta r_2 & -\delta r_1 & 1 \end{bmatrix} \tag{10}$$

However, Equation 10 is not always a $SO(3)$ matrix. As a result, we need convert δR to a quaternion, normalize the quaternion, and then convert the quaternion back to δR. Dividing Equation 7

by λ_a and substituting, we get the following:

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \begin{bmatrix} \frac{r_1^T \begin{bmatrix} x_a & y_a & 1 \end{bmatrix}^T}{r_3^T \begin{bmatrix} x_a & y_a & 1 \end{bmatrix}^T} \\ \frac{r_2^T \begin{bmatrix} x_a & y_a & 1 \end{bmatrix}^T}{r_3^T \begin{bmatrix} x_a & y_a & 1 \end{bmatrix}^T} \end{bmatrix} = g\left({}^{C_b}R^{C_a}, \begin{bmatrix} x_a \\ y_a \end{bmatrix} \right) \tag{11}$$

Equation 11 is the self-calibration constraint for 1 point correspondence.

We can stack Equation 11 for multiple point correspondences to form a residual vector. We can augment the residual vector with constraints on variables. Therefore, the residual vector is of the form:

$$r = \begin{bmatrix} \begin{bmatrix} x_{b1} \\ y_{b1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_{bn} \\ y_{bn} \end{bmatrix} \\ \alpha_1 c_1(\beta) \\ \vdots \\ \alpha_m c_m(\beta) \end{bmatrix} - \begin{bmatrix} g\left({}^{C_b}R^{C_a}, \begin{bmatrix} x_{a1} \\ y_{a1} \end{bmatrix} \right) \\ \vdots \\ g\left({}^{C_b}R^{C_a}, \begin{bmatrix} x_{an} \\ y_{an} \end{bmatrix} \right) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{12}$$

for n correspondences and m constraints. The vector

$$\beta = \begin{bmatrix} f_x & f_y & s & u_0 & v_0 & k_1 & k_2 & k_3 & p_1 & p_2 & \delta r_1 & \delta r_2 & \delta r_3 \end{bmatrix}^T \tag{13}$$

is what we are trying to optimize. Examples of constraints include zero-skew

$$c(\beta) = -s$$

or equal focal lengths

$$c(\beta) = 1 - \frac{f_x}{f_y}$$

The parameter α is a weighting factor to control the relative importance between self-calibration constraint and equality constraint. In summary, the optimization problem we are trying to solve is:

$$\beta^* = \arg\min_{\beta} \frac{1}{2} \|r\|_2^2 \tag{14}$$

which is a nonlinear least squares problem. Note that if more than 2 images are available, Equation 12 can be augmented with self-calibration constraints for any combinations of 2 images.

VI. Levenberg-Marquardt Optimization

The Levenberg-Marquardt (LM) optimization algorithm will be implemented to solve Equation 14 [16]. To start, we define the Jacobian as:

$$J(\beta) = \frac{\partial r}{\partial \beta} \tag{15}$$

The Jacobian can be calculated using successive chain rules. We define helper functions:

$$F(\beta) = \frac{1}{2}\|r(\beta)\|_2^2 \qquad L(\beta, h) = F(\beta) + h^T J(\beta)^T r(\beta) + \frac{1}{2} h^T J(\beta)^T J(\beta) h$$

where h is a step taken for a LM iteration and $L(\beta, h)$ is the first order approximation of $F(x+h)$. The full algorithm is then:

Algorithm 1 Levenberg-Marquardt Algorithm with Attitude Correction and Constraints

Require: β_0, k_{max}, τ, ε_1, and ε_2
$\beta = \beta_0$; $k = 0$; $\nu = 2$
$A = J(\beta)^T J(\beta)$; $g = J(\beta)^T r(\beta)$
found = $\|g\|_\infty \leq \varepsilon_1$
$\mu = \tau \max(A(i,i))$
while not found and $k < k_{max}$ **do**
    $k = k + 1$
    solve $(A + \mu I_{3x3})h = -g$ for h
    if $\|h\|_2 \leq \varepsilon_2(\|\beta\|_2 + \varepsilon_2)$ **then**
        found = true
    else
        $\beta_{new} = \beta + h$
        $\rho = \frac{F(\beta) - F(\beta_{new})}{L(\beta,0) - L(\beta,h)}$
        if $\rho > 0$ **then**
            $\beta = \beta_{new}$
            turn δR into SO(3), update R, reset δR
            $A = J(\beta)^T J(\beta)$; $g = J(\beta)^T r(\beta)$
            found = $\|g\|_\infty \leq \varepsilon_1$
            $\mu = \mu \max(1/3, 1 - (2\rho - 1)^3)$; $\nu = 2$
        else
            $\mu = \mu\nu$; $\nu = 2\nu$
        end if
    end if
end while
return β, R

μ is the damping parameter that interpolates the algorithm between gradient descent and Gauss-Newton iteration. μ is controlled by the gain ratio ρ. If ρ is large, $L(\beta, h)$ is a good approximation

of $F(\beta)$ and Gauss-Newton is preferred. If ρ is small, then gradient descent is preferred [16].

VII. RESULTS

First, the algorithm is examined for zero noise case. 20 random inertial points are generated. Using the method in Figure 2, the observed image positions in pixel coordinates are generated for each orientation with

$$\beta_{truth} = \begin{bmatrix} 2714.286 & 2714.286 & 0 & 1640 & 1232 & 0.3 & 0.2 & 0 & 0.1 & -0.1 & 0 & 0 & 0 \end{bmatrix}^T$$

Signification deviation is then introduced to β_0:

$$\beta_0 = \begin{bmatrix} 2000 & 2000 & 0.2 & 1500 & 1100 & 0.5 & 0 & 0.15 & 0.3 & -0.15 & 0 & 0 & 0 \end{bmatrix}^T$$

A large initial error in the rotation matrix is introduced via quaternion error

$$\delta q_0 = \begin{bmatrix} -0.0696 \\ -0.0015 \\ 0.4417 \\ 0.8945 \end{bmatrix}$$

where

$$\delta q^{(k)} = q^{(k)} \otimes q_{truth}^{-1}$$

for iteration k. The symbol $\otimes$ represents quaternion multiplication. The attitude error is zero when $\delta q = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$, the identity quaternion.

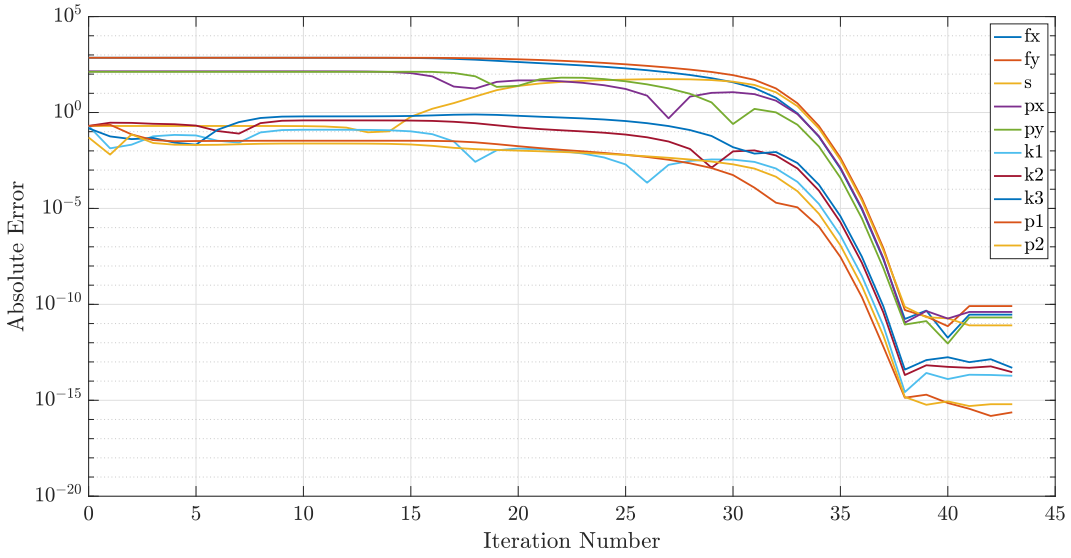

Figure 3: 20 Point Correspondences, No Noise

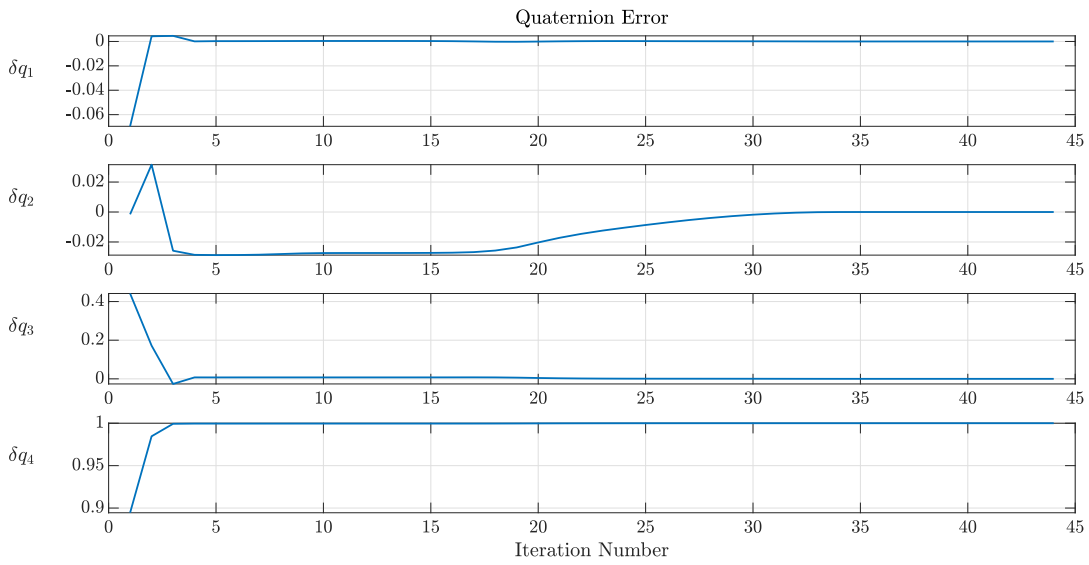

Figure 4: Quaternion Error, No Noise

As shown in Figure 3, the algorithm converges to a solution in 44 iterations. The solution is accurate to within 9 decimal places. From Figure 4, the attitude error converges to the identity quaternion.

Image noise is introduced assuming additive white Gaussian noise $N(0_{3x1}, \sigma^2)$.

σ	f_x	f_y	s	u_0	v_0	k_1	k_2	k_3	p_1	p_2
0.1	2714.118	2714.468	0.065	1640.182	1232.121	0.299	0.207	-0.018	0.1	-0.1
0.25	2715.602	2717.81	2.36	1641.168	1233.145	0.3	0.194	0.0244	0.1	-0.1
0.5	2707.425	2715.095	1.079	1645.708	1239.626	0.297	0.253	-0.15	0.1	-0.1
1	2706.548	2713.497	14.066	1649.303	1238.870	0.308	0.16	0.093	0.099	-0.1

Table 1: Solution with additive white Gaussian noise

As shown in Table 1, the errors in parameters increase gradually with increasing image noise. However, the skew factor error increases much more rapidly compared to other parameters. Since the skew factor is likely to be unchanged, to mitigate this effect, constraints on the the skew factor can be introduced in the algorithm.

VIII. Conclusion

An on-orbit camera self-calibration approach using relative point correspondences is introduced. The approach uses Levenberg-Marquardt nonlinear least squares algorithm to find an optimal set

of parameters that minimizes the self-calibration residuals as well as optional known constraints on the parameters. Through sensitivity analysis, the algorithm performance degrades gradually with increasing image noise. Future work includes analyzing how errors in calibration parameters propagate to optical navigation state estimation errors.

IX. ACKNOWLEDGMENTS

The authors would like to thank the NASA Cube Quest Challenge for selecting the Cislunar Explorers for a launch spot on Exploration Mission-1. We would also like to thank the National Space Society (NSS) for continued monetary and technical support.

REFERENCES

[1] R. I. Hartley, "Self-calibration of stationary cameras," *International Journal of Computer Vision*, vol. 22, pp. 5–23, Feb 1997.

[2] M. Wang, Y. Cheng, B. Yang, J. Shuying, and H. Su, "On-orbit calibration approach for optical navigation camera in deep space exploration," vol. 24, p. 5536, 03 2016.

[3] J. A. Christian, L. Benhacine, and J. Hikes, "Geometric calibration of the orion optical navigation camera using star field images," *Advances in the Astronautical Sciences*, vol. 157, pp. 769–780, 2016.

[4] D. Mortari, F. De Dilectis, and C. D'Souza, "Image processing of illuminated ellipsoid," vol. 150, pp. 2245–2262, 01 2014.

[5] G. E. Lightsey and J. A. Christian, "Onboard image-processing algorithm for a spacecraft optical navigation sensor system," *Journal of Spacecraft and Rockets*, vol. 49, pp. 337–352, 2018/05/28 2012.

[6] J. A. Christian, "Optical navigation using planet's centroid and apparent diameter in image," vol. 38, pp. 192–204, 02 2015.

[7] V. H. Adams and M. A. Peck, *Interplanetary Optical Navigation*. American Institute of Aeronautics and Astronautics, 2018/05/28 2016.

[8] V. H. Adams and M. A. Peck, *Lost in Space and Time*. American Institute of Aeronautics and Astronautics, 2018/05/28 2017.

[9] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems, Second Edition (Chapman & Hall/CRC Applied Mathematics & Nonlinear Science)*. Chapman & Hall/CRC, 2nd ed., 2011.

[10] G. P. Stein, "Lens distortion calibration using point correspondences," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 602–608, Jun 1997.

[11] B. K. P. Horn, "Relative orientation revisited," *J. Opt. Soc. Am. A*, vol. 8, pp. 1630–1638, Oct 1991.

[12] J. A. Christian, "Accurate planetary limb localization for image-based spacecraft navigation," vol. 54, pp. 708–730, 05 2017.

[13] R. Hu and Q. Ji, "Camera self-calibration from ellipse correspondences," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3, pp. 2191–2196 vol.3, 2001.

[14] G. N. Holt, C. N. D'Souza, and D. W. Saley, *Orion Optical Navigation Progress Toward Exploration Mission 1*. American Institute of Aeronautics and Astronautics, 2018/05/28 2018.

[15] M. Paluszek, M. Littman, and J. Mueller, *Optical Navigation System*. American Institute of Aeronautics and Astronautics, 2018/05/28 2010.

[16] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems (2nd ed.)," p. 60, 01 2004.

[17] Raspberry Pi Foundation, *Raspberry Pi Camera Module v2*, 4 2016.

[18] R. Szeliski, *Computer Vision: Algorithms and Applications*. Berlin, Heidelberg: Springer-Verlag, 1st ed., 2010.